

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-073576

(43)Date of publication of application : 12.03.2002

(51)Int.Cl. G06F 15/16  
 G06F 9/46  
 G06F 11/00  
 G06F 15/177

(21)Application number : 2000-262916

(71)Applicant : TOSHIBA CORP

(22)Date of filing : 31.08.2000

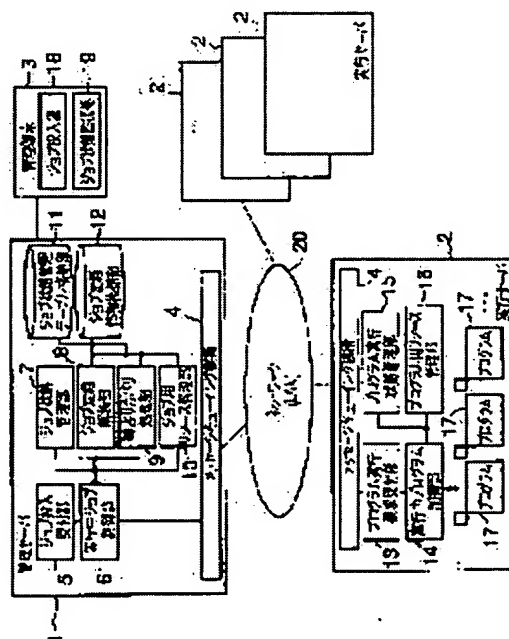
(72)Inventor : MATSUO AKIRA

## (54) BATCH JOB CONTROL SYSTEM

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a batch job control system which can easily and surely perform a series of processings on a job constituted of a plurality of processing programs on a plurality of computers which are mutually connected through a network.

**SOLUTION:** The batch job control system is provided with a managing server 1 managing the performing order of the respective processing programs in the job and the computer becoming a performing destination, a plurality of performing servers 2 performing the respective processing programs in the job and a managing terminal 3 connected to the managing server 1. The managing server 1 is connected to each of the performing servers 2 through the network 20. The managing server 1 and the performing servers 2 respectively have message queuing mechanisms 4 for performing asynchronous message communication. The respective processing programs are associated by transferring the performing request of the respective processing programs in the job as a message between the managing server 1 and the performing servers 2 through the message queuing mechanisms 4.



\* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

1.This document has been translated by computer. So the translation may not reflect the original precisely.

2.\*\*\*\* shows the word which can not be translated.

3.In the drawings, any words are not translated.

---

## CLAIMS

---

[Claim(s)]

[Claim 1]Have the following and said administrative computer and said each running calculation machine have a message queuing means for performing asynchronous message communication, respectively, . It is characterized by making said each processing program cooperate by exchanging an execution demand of each processing program in a job as a message via said message queuing means between said administrative computer and said each running calculation machine. A batch job control system which controls execution of a job which consists of two or more processing programs, makes each processing program concerned cooperate, and performs a series of processings.

An administrative computer which manages a computer used as execution sequence of each processing program in a job, and a real destination.

Two or more running calculation machines which are connected to said administrative computer and execute each processing program in a job.

[Claim 2]The batch job control system comprising according to claim 1:

A job supply receiving means in which said administrative computer receives an execution demand of a job.

A job definition information storing means which stores job definition information about a computer used as execution sequence of each processing program in a job, and a real destination.

A job definition analysis means to determine a computer used as execution sequence of each processing program in the job concerned, and a real destination about a job which analyzed job definition information stored in said job definition information storing means, and was received by said job supply receiving means.

Execution of a job received by said job supply receiving means is controlled, Based on execution sequence determined by said job definition analysis means, it has an executing job control means which transmits an execution demand of a processing program applicable to an applicable running calculation machine as a message, A program execution demand receiving means in which said each running calculation machine receives a message of an execution demand of each processing program in a job, It is a programmed control means during execution which executes a processing program corresponding to an execution demand received by said program execution demand receiving means, and transmits the execution terminating notice as a message to said administrative computer.

[Claim 3]The batch job control system according to claim 2, wherein said administrative computer has further a job state management tool which manages stages of progress of an active job and said each running calculation machine has further a program execution status management means to manage stages of progress of a processing program under execution.

[Claim 4]Said job definition information storing means of said administrative computer, As said a part

of job definition information, store further information about a recovering method of each processing program in a job or a job, and said administrative computer, When an obstacle occurs during execution of a job. The batch job control system according to claim 2 having further an obstacle recovery processing means to perform a restoration process of each processing program in a job or a job based on information about said recovering method stored in said job definition information storing means.

[Claim 5] Said job definition information storing means of said administrative computer, Store further information about a computer resource needed as said a part of job definition information when performing a job, and said administrative computer, The batch job control system according to claim 2 having further a resource management means for jobs to manage a computer resource needed when performing a job based on information about said computer resource stored in said job definition information storing means per job.

[Claim 6] Said job definition information storing means of said administrative computer, Store further information about a computer resource needed as said a part of job definition information when executing each processing program in a job, and said administrative computer, Transmit with an execution demand of each processing program in a job to said each running calculation machine, and information about said computer resource said each running calculation machine, Having further a resource management means for a program to manage a computer resource needed when executing each processing program in a job based on information about said computer resource transmitted from said administrative computer per processing program. The batch job control system according to claim 2 by which it is characterized.

[Claim 7] In a batch job control system which controls execution of a job which consists of two or more processing programs, makes each processing program concerned cooperate, and performs a series of processings, Have two or more running calculation machines which execute each processing program in a job, and said each running calculation machine has a message queuing means for performing asynchronous message communication, respectively, A batch job control system making said each processing program cooperate by exchanging an execution demand of a job as a message via said message queuing means between said each running calculation machine.

[Claim 8] The batch job control system comprising according to claim 7:

A job execution request receiving means which receives job definition information about a computer by which said each running calculation machine serves as execution sequence of each processing program in a job, and a real destination as an execution demand of a job, and information about stages of progress of an active job.

A job definition analysis means to determine a computer which analyzes said job definition information and information about stages of progress of said active job, and serves as execution sequence of a processing program which is not executed in a job, and a real destination about a job received by said job execution request receiving means.

Execution of a job received by said job supply receiving means is controlled, Based on execution sequence of a processing program which is not executed in a job determined by said job definition analysis means, An executing job control means which transmits said job definition information and information about stages of progress of said active job as a message as an execution demand of a job which contains an unperformed processing program to an applicable running calculation machine. It is a programmed control means during execution which executes a processing program which should be executed with the running calculation machine concerned among processing programs in a job received by said job execution request receiving means.

[Claim 9] The batch job control system according to claim 8, wherein said each running calculation machine has further a job state management tool which manages stages of progress of an active job, and a program execution status management means to manage stages of progress of a processing program under execution.

[Claim 10] Said job definition information, including further information about a recovering method of each processing program in a job or a job said each running calculation machine, The batch job control system according to claim 8 having further an obstacle recovery processing means to perform a restoration process of each processing program in a job or a job based on information about said recovering method included in said job definition information when an obstacle occurs during execution of a job.

[Claim 11] Said job definition information, including further information about a computer resource needed when performing a job said each running calculation machine, The batch job control system according to claim 8 having further a resource management means for jobs to manage a computer resource needed when performing a job based on information about said computer resource contained in said job definition information per job.

[Claim 12] Said job definition information, including further information about a computer resource needed when executing each processing program in a job said each running calculation machine, Having further a resource management means for a program to manage a computer resource needed when executing each processing program in a job based on information about said computer resource contained in said job definition information per processing program. The batch job control system according to claim 8 by which it is characterized.

---

[Translation done.]

\* NOTICES \*

JP0 and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention relates to the batch job control system which controls execution of the job which consists of two or more processing programs, and relates to the batch job control system which makes each processing program in a job cooperate among two or more computers mutually connected via division and a network, and performs a series of processings.

[0002]

[Description of the Prior Art]The batch job control system which carries out sequential execution of two or more processing programs in a job on a single computer as a system which controls from the former execution of the job which consists of two or more processing programs is known.

[0003]By the way, to use widely the distributing system to which two or more computers of each other were connected with the spread of network, to make each processing program in a job cooperate on such a distributing system, and to perform a series of processings is desired in recent years.

[0004]

[Problem(s) to be Solved by the Invention]However, in a distributing system which was mentioned above, since two or more computers of each other are connected via the network, there is a problem that it is difficult to make several processing programs with which processing time differs from processing timing etc. among these computers cooperate.

[0005]In a distributing system which was mentioned above, under open environment, several computers by which plat forms (OS etc.) differ are intermingled, and are used in many cases, and there is a problem that it is difficult to exchange data appropriately among these computers.

[0006]It is necessary to perform processing and control of a lot of data among two or more computers in a distributing system which was mentioned above, and the obstacle of a network, a computer, etc. is received, There is a problem that it is difficult to guarantee processing of the data exchanged between computers and a control flow, and to maintain reliability.

[0007]This invention is made in consideration of such a point, and is a thing.

It is providing the batch job control system which can perform easily and certainly processing of a series of the job which consists of two or more processing programs on two or more computers mutually connected via the purpose.

[0008]

[Means for Solving the Problem]In a batch job control system which this invention controls [ batch job control system ] execution of a job which consists of two or more processing programs as the 1st solving means, makes each processing program concerned cooperate, and performs a series of processings, An administrative computer which manages a computer used as execution sequence of each processing program in a job, and a real destination, It is connected to said administrative

computer and has two or more running calculation machines which execute each processing program in a job, Said administrative computer and said each running calculation machine have a message queuing means for performing asynchronous message communication, respectively, A batch job control system making said each processing program cooperate is provided by exchanging an execution demand of each processing program in a job as a message via said message queuing means between said administrative computer and said each running calculation machine.

[0009]In the 1st solving means mentioned above, said administrative computer, A job definition information storing means which stores job definition information about a computer used as a job supply receiving means which receives an execution demand of a job, and execution sequence of each processing program in a job and a real destination, Job definition information stored in said job definition information storing means is analyzed, A job definition analysis means to determine a computer used as execution sequence of each processing program in the job concerned, and a real destination about a job received by said job supply receiving means, Execution of a job received by said job supply receiving means is controlled, Based on execution sequence determined by said job definition analysis means, it has an executing job control means which transmits an execution demand of a processing program applicable to an applicable running calculation machine as a message, A program execution demand receiving means in which said each running calculation machine receives a message of an execution demand of each processing program in a job, It is preferred to have a programmed control means during execution which executes a processing program corresponding to an execution demand received by said program execution demand receiving means, and transmits the execution terminating notice as a message to said administrative computer.

[0010]In the 1st solving means mentioned above, said administrative computer has further a job state management tool which manages stages of progress of an active job, and, as for said each running calculation machine, it is preferred to have further a program execution status management means to manage stages of progress of a processing program under execution. Said job definition information storing means of said administrative computer, As said a part of job definition information, store further information about a recovering method of each processing program in a job or a job, and said administrative computer, When an obstacle occurs during execution of a job, it is preferred to have further an obstacle recovery processing means to perform a restoration process of each processing program in a job or a job based on information about said recovering method stored in said job definition information storing means. Said job definition information storing means of said administrative computer, Store further information about a computer resource needed as said a part of job definition information when performing a job, and said administrative computer, It is preferred to have further a resource management means for jobs to manage a computer resource needed when performing a job based on information about said computer resource stored in said job definition information storing means per job. Further again said job definition information storing means of said administrative computer, Store further information about a computer resource needed as said a part of job definition information when executing each processing program in a job, and said administrative computer, Transmit with an execution demand of each processing program in a job to said each running calculation machine, and information about said computer resource said each running calculation machine, It is preferred to have further a resource management means for a program to manage a computer resource needed when executing each processing program in a job based on information about said computer resource transmitted from said administrative computer per processing program.

[0011]In a batch job control system which this invention controls [ batch job control system ] execution of a job which consists of two or more processing programs as the 2nd solving means, makes each processing program concerned cooperate, and performs a series of processings, Have two or more running calculation machines which execute each processing program in a job, and said each running calculation machine has a message queuing means for performing asynchronous

message communication, respectively, A batch job control system making said each processing program cooperate is provided by exchanging an execution demand of a job as a message via said message queuing means between said each running calculation machine.

[0012]In the 2nd solving means mentioned above, said each running calculation machine, A job execution request receiving means which receives job definition information about a computer used as execution sequence of each processing program in a job, and a real destination, and information about stages of progress of an active job as an execution demand of a job, About a job received by said job execution request receiving means. A job definition analysis means to determine a computer which analyzes said job definition information and information about stages of progress of said active job, and serves as execution sequence of a processing program which is not executed in a job, and a real destination, Execution of a job received by said job supply receiving means is controlled, Based on execution sequence of a processing program which is not executed in a job determined by said job definition analysis means, An executing job control means which transmits said job definition information and information about stages of progress of said active job as a message as an execution demand of a job which contains an unperformed processing program to an applicable running calculation machine, It is preferred to have a programmed control means during execution which executes a processing program which should be executed with the running calculation machine concerned among processing programs in a job received by said job execution request receiving means.

[0013]As for said each running calculation machine, in the 2nd solving means mentioned above, it is preferred to have further a job state management tool which manages stages of progress of an active job, and a program execution status management means to manage stages of progress of a processing program under execution. Said job definition information, including further information about a recovering method of each processing program in a job or a job said each running calculation machine, When an obstacle occurs during execution of a job, it is preferred to have further an obstacle recovery processing means to perform a restoration process of each processing program in a job or a job based on information about said recovering method included in said job definition information. Said job definition information, including further information about a computer resource needed when performing a job said each running calculation machine, It is preferred to have further a resource management means for jobs to manage a computer resource needed when performing a job based on information about said computer resource contained in said job definition information per job. Further again said job definition information, including further information about a computer resource needed when executing each processing program in a job said each running calculation machine, It is preferred to have further a resource management means for a program to manage a computer resource needed when executing each processing program in a job based on information about said computer resource contained in said job definition information per processing program.

[0014]According to the 1st and 2nd solving means of this invention, the message queuing mechanism 4 for performing asynchronous message communication to each of two or more computers is formed, Since each processing program is made to cooperate by exchanging an execution demand of each processing program in a job, or an execution demand of a job as a message via the message queuing mechanism 4, Processing of a series of a job which consists of two or more processing programs on two or more computers mutually connected via a network can be performed easily and certainly.

[0015]

[Embodiment of the Invention]Hereafter, an embodiment of the invention is described with reference to drawings.

[0016]The 1st embodiment drawing 1 thru/or drawing 5 is a figure for describing a 1st embodiment of the batch job control system by this invention.

[0017]First, drawing 1 explains the entire configuration of the batch job control system concerning a

1st embodiment of this invention. As shown in drawing 1, a batch job control system controls execution of the job which consists of two or more batch processing programs (only henceforth a "processing program"), makes each processing program concerned cooperate, and performs a series of processings. A batch job control system is provided with the following.

The managing server (administrative computer) 1 which manages the computer used as the execution sequence of each processing program in a job, and a real destination.

Two or more execution servers (running calculation machine) 2 which are connected to the managing server 1 and execute each processing program in a job.

The administration terminal 3 connected to the managing server 1.

[0018]In drawing 1, the managing server 1 and each execution server 2 are connected via the networks 20, such as LAN. The managing server 1 and each execution server 2 have the message queuing mechanism 4 for performing respectively asynchronous message communication. Each processing program is made to cooperate by exchanging the execution demand of each processing program in a job as a message via the message queuing mechanism 4 between the managing server 1 and each execution server 2.

[0019]Here, the managing server 1 is provided with the following.

The job supply reception part 5 which receives an execution demand of a job.

The job definition information storing part 12 which stores the job definition information about the computer used as the execution sequence of each processing program in a job, and a real destination.

To the information about the computer which becomes job definition information with the execution sequence of each processing program in a job, and a real destination, in addition, the information about the recovering method of each processing program in a job or a job, The information about the computer resource needed when executing each processing program in a job or a job is stored. The details of job definition information are mentioned later.

[0020]The managing server 1 is provided with the following.

Job definition analyzing parts 8 which determine the computer used as the execution sequence of each processing program in the job concerned, and a real destination about the job which analyzed the job definition information stored in the job definition information storing part 12, and was received by the job supply reception part 5.

The executing job control section 6 which transmits as a message the execution demand of the processing program which corresponds to the applicable execution server 2 based on the execution sequence which controlled execution of the job received by the job supply reception part 5, and was determined by the job definition analyzing parts 8.

Here the executing job control section 6 of the managing server 1, After transmitting the execution demand of a processing program as a message to the execution server 2, when waiting and the execution terminating notice concerned are received, the execution terminating notice of a processing program from the execution server 2, The execution demand of the following processing program is transmitted as a message, and the same processing is repeated until execution of the last processing program is completed. The message transmitted from the executing job control section 6 of the managing server 1 is transmitted to each execution server 2 via the message queuing mechanism 4 of the managing server 1.

[0021]The managing server 1 is provided with the following.

The job state management table storage 11 which stores the job state management table showing the stages of progress of an active job.

The job state Management Department 7 which manages the stages of progress of an active job using the job state management table stored in the job state management table storage 11.

The information which processing program in a job is executed by the job state management table on which computer now etc. The status information of the processing program in the already ended



job or a job is stored, and it is updated by the job state Management Department 7 based on the demand from the executing job control section 6. The details of a job state management table are mentioned later.

[0022]The managing server 1 is provided with the following further again.

The obstacle recovery processing part 9 which performs the restoration process of each processing program in a job or a job based on the information about the recovering method stored in the job definition information storing part 12 when an obstacle occurred during execution of a job.

The resource management department 10 for jobs which manages the computer resource needed when performing a job based on the information about the computer resource stored in the job definition information storing part 12 per job.

The resource management department 10 for jobs is what manages various computer resources, such as a memory, CPU, and disk storage capacity (reservation and opening), When starting execution of the job by which the execution demand was received, a required computer resource is secured based on the information about the computer resource stored in the job definition information storing part 12.

[0023]Each execution server 2 is provided with the following.

The program execution demand accepting part 13 which receives the message of an execution demand of each processing program in the job transmitted from the executing job control section 6 of the managing server 1.

It is the program control part 14 during the execution which executes the processing program corresponding to the execution demand received by the program execution demand accepting part 13, and transmits the execution terminating notice as a message to the managing server 1.

During execution, the program control part 14 performs the surveillance from starting of the processing program corresponding to the execution demand received by the program execution demand accepting part 13 to an end, and transmits the termination status to the executing job control section 6 of the managing server 1. The message transmitted from the program control part 14 during execution of each execution server 2 is transmitted to the managing server 1 via the message queuing mechanism 4 of each execution server 2.

[0024]Each execution server 2 is provided with the following.

The program execution status management department 15 which manages stages of progress of the processing program under execution, such as which processing program is simultaneously executed how many on the execution server 2.

The resource management department 16 for a program which manages the computer resource needed when executing each processing program in a job per processing program.

The resource management department 16 for a program like the resource management department 10 for jobs of the managing server 1, It is what manages various computer resources, such as a memory, CPU, and disk storage capacity (reservation and opening), When starting execution of a processing program on each execution server 2, a required computer resource is secured based on the information about the computer resource handed over from the managing server 1. Two or more processing programs 17 actually started are installed in the execution server 2.

[0025]The managing server 1 and the message queuing mechanism 4 of each execution server 2, Queuing of the message exchanged between the managing server 1 and each execution server 2 is carried out, a message is asynchronously exchanged between the managing server 1 and each execution server 2, and it guarantees that a message arrives certainly to the execution server 2 which serves as a transmission destination. Namely, the message queuing mechanism 4, When a message cannot transmit according to the obstacle of the network 20, a computer (the managing server 1 and the execution server 2), etc., queuing of the message is carried out, and when it will be in the state which can communicate, it has an automatic retry function which transmits a message automatically. For this reason, the managing server 1 or each execution server 2, Processing of the managing server 1 or each execution server 2 can be continued without waiting for the check of

arrival of the message to the computer used as the partner point, after handing over a message from the program control part 14 to the message queuing mechanism 4 during the executing job control section 6 or execution.

[0026]The administration terminal 3 is provided with the following.

The job supply part 18 which is the computer for users (administrator) connected to the managing server 1, and supplies a job to the time set up beforehand to the managing server 1 according to a user's directions automatically (execution demand).

The job state Monitoring Department 19 which supervises the stages of progress of the active job managed with the managing server 1 with reference to the job state management table stored in the job state management table storage 11.

[0027]Next, drawing 1 and drawing 2 explain an operation of a 1st embodiment of this invention which consists of such composition. Drawing 2 is a flow chart for explaining the flow of processing of the whole batch job control system shown in drawing 1.

[0028]First, a user creates job definition information for every job used as the candidate for execution, and stores in the job definition information storing part 12. Here, the work which creates job definition information can be done using the definition tool for exclusive use provided with GUI etc. in administration terminal 3 superiors. As job definition information, about each processing program in a job. The information about the recovering method of each processing program in a job or a job besides the information about the computer used as the program name, execution sequence, and a real destination, the information about the computer resource needed when executing each processing program in a job or a job, etc. are stored.

[0029]Drawing 3 is a figure showing an example of the job definition information stored in the job definition information storing part 12.

[0030]As shown in drawing 3, job definition information 12' consists of three files, the flow definition master file 12a, the execution sequence definition file 12b, and the processing program attribute definition file 12c.

[0031]Among these, the flow definition master file 12a, It is a master file of job definition information, and the file name ("FILE=F01.net") of the execution sequence definition file 12b besides an identification number ("ID=00000001") and a jobname "NAME= business A" is described.

[0032]The execution sequence definition file 12b defines the execution sequence of each processing program in a job, and the identification number of each processing program and two or more sets of identification numbers of the processing program preceded with it are described.

[0033]In the execution sequence definition file 12b shown in drawing 3. After executing the processing program of the identification numbers 0010 and 0011 one by one by making the processing program of the identification number 0010 into a head, The example of description in the case of it being parallel and executing the processing program of the identification numbers 0012 and 0013, waiting the end of the processing program of the identification numbers 0012 and 0013, and finally, executing the processing program of the identification number 0014 is shown.

[0034]Drawing 4 is a mimetic diagram showing the flow of the job according to the execution sequence definition file 12b shown in drawing 3. As shown in drawing 4, the flow of a job like the processing program of the (1) identification numbers 0010 and 0011, Like the processing program of the type (one by one mold) which waits for the end of the processing program to precede and starts execution of the following processing program, and the (2) identification numbers 0012 and 0013, It can express as combination of the type (joined type) which waits the end of two or more processing programs to precede like the type (branch-type) which is parallel and executes two or more processing programs simultaneously, and the processing program of the (3) identification number 0014, and starts execution of the following processing program.

[0035]The processing program attribute definition file 12c defines the attribution information of each processing program in a job, Identification numbers ("ID=0010" etc.), a program name ("NAME=

processing A" etc.), The computer name ("HOST=SERVER-A" etc.) used as the paths ("PATH=C:/... /p10.bat" etc.) of a program, and a real destination, etc. The information (not shown) about the computer resource needed when executing each processing program in the names ("RECOVERY= recovery A" etc.) of the recovering method of each processing program in a job or a job and a job, or a job is described.

[0036]After creating job definition information as mentioned above, a user supplies a job using the job supply part 18 of the administration terminal 3 (execution demand). Here, the job which can perform an execution demand is only a job by which job definition information is beforehand stored in the job definition information storing part 12. A user's directions perform an execution demand of a job, and also the starting schedule (job used as the time of an opening day and the candidate for a start) is set up beforehand, and it can be automatically carried out to the set-up time.

[0037]As shown in drawing 2, according to a user's directions, an execution demand of the job automatically taken out from the job supply part 18 to the time set up beforehand is received by the job supply reception part 5 of the managing server 1 (Step 101), and is handed over to the executing job control section 6. The executing job control section 6 asks to the job definition analyzing parts 8, The job definition information (the flow attribute definition master file 12a and the execution sequence definition file 12b of drawing 3) stored in the job definition information storing part 12 is read (Step 102), While extracting the processing program of the head of a job from which the execution demand was advanced, the computer used as the real destination of the processing program is determined (Step 103).

[0038]Then, the executing job control section 6 makes the execution demand of an applicable processing program a message, it hands over to the message queuing mechanism 4, and a message is transmitted to it to the applicable execution server 2 (Step 104). The message queuing mechanism 4 carries out queuing of the message handed over from the executing job control section 6, and exchanges a message asynchronously between the message queuing mechanisms 4 of the applicable execution server 2. The message queuing mechanism 4 has an automatic retry function here, In the case where an obstacle occurs to the network 20 and a message cannot transmit to it, the case where the execution server 2 which serves as a transmission destination is downed, and a message cannot transmit, etc., After an obstacle occurs before restoring from an obstacle, a retry is automatically tried with a constant interval, and when it will be in the state which can communicate, a message is transmitted automatically. When a flow branches from one processing program in a job to two or more processing programs, the multiple address of the message is carried out from the managing server 1 to two or more execution servers 2. The message exchanged between the managing server 1 and the execution server 2, It is equivalent to the processing program attribute file 12c shown in drawing 3, and the path of a program name and a program, the computer name used as a real destination, the information on an option, including information about the name and computer resource of a recovering method, etc., etc. are included.

[0039]Here, the message of an execution demand of the processing program transmitted from the executing job control section 6 of the managing server 1 is received by the program execution demand accepting part 13 of the applicable execution server 2 (Step 201). Then, the message of an execution demand of the processing program received by doing in this way is handed over during execution to the program control part 14, and the applicable processing program 17 is started according to the information included in the message concerned (Step 202). In the program control part 14, it is also possible during execution to start several different processing programs 17 simultaneously, and two or more threads are started in that case corresponding to each processing program.

[0040]Thus, if a processing program is started, the program control part 14 will register the processing program under execution to the program execution status management department 15 during execution (Step 203). Here, the program execution status management department 15 manages how many the processing program is executed simultaneously, and controls it not to

perform many big programs of a processing load simultaneously.

[0041]During execution, the program control part 14 performs the registration deletion request of a processing program to the program execution status management department 15 in a similar manner, after the started execution of a processing program is completed (Step 204). During execution, the program control part 14 acquires the termination status of the ended processing program, and transmits the execution terminating notice of a processing program as a message with the termination status to the executing job control section 6 of the managing server 1 (Step 205). This message is transmitted to the managing server 1 via the message queuing mechanism 4 of the execution server 2.

[0042]Thus, the executing job control section 6 of the managing server 1 receives the execution terminating notice transmitted from the execution server 2 (Step 105). The executing job control section 6 hands over the execution terminating notice which received to the job state Management Department 7 with termination status, and updates the job state management table stored in the job state management table storage 11 (Step 106).

[0043]Drawing 5 is a figure showing an example of the job state management table stored in the job state management table storage 11.

[0044]As shown in drawing 5, job state management table 11', the information (job ID, a task name, an input date, and an injection user.) about the stages of progress of a job The information (at program ID, a processing name, an input date, execution start time, the time of an execution end date a running calculation machine, termination status) about the stages of progress of each processing program in a job besides a job step is included during the present status and execution at the time of an end date.

[0045]After execution of one processing program is completed, the executing job control section 6 asks to the job definition analyzing parts 8 again, and judges whether the following processing program exists in the job definition information on the job concerned (Step 107).

[0046]Here, when the following processing program exists, while extracting the following processing program concerned, the computer used as the real destination of the processing program is determined (Step 108), and processing of Steps 104 thru/or 107 mentioned above is repeated. When a flow joins from two or more processing programs to one processing program and waiting and all the execution terminating notices gather until all of the required execution terminating notice of two or more processing programs to precede gather, the execution demand of the following processing program is transmitted as a message.

[0047]One job is completed when the processing program of the last in the job concerned is completed (Step 109).

[0048]When an obstacle occurs during execution of the job accompanied by such a series of processings here, The obstacle recovery processing part 9 performs the restoration processes (the rerun of a processing program, a stop, the reclosing of a job, etc.) of each processing program in a job or a job based on the information about the recovering method stored in the job definition information storing part 12. A user can start the obstacle recovery processing part 9.

[0049]As an obstacle generated during execution of a job, for example, the case where the processing program under execution caused the error by a certain cause, and is completed, Although the job interrupted on the way according to the obstacle of the network 20, a computer (the managing server 1 and the execution server 2), etc. repeats an automatic retry, it may stagnate over a long time. The recovering method in such a case changes with kinds of job, and can take the method of reperforming again from the processing program which caused the error, and the method of skipping the processing program which caused the error and resuming as a method of resuming a job. The method of reclosing a job and reperforming from the first processing program, the method of terminating a job or a processing program compulsorily [ when an error occurs ], etc. can be taken. The recovering method taken to the processing program in each job or a job is described by the job definition information, including processing program attribute definition file 12c etc., stored in

the job definition information storing part 12.

[0050]In resuming a job, here the obstacle recovery processing part 9, The information about the stages of progress of each processing program in a job is acquired from the job state management table stored in the job state management table storage 11, After determining the processing program to resume in accordance with the recovering method described by job definition information, the resumption demand of a processing program, etc. are given to the executing job control section 6. At this time, the obstacle recovery processing part 9 deletes all the related messages left behind to the message queuing mechanism 4. In reclosing a job, the obstacle recovery processing part 9 deletes all the related messages left behind to the message queuing mechanism 4, and directs a reclosing to the job supply reception part 5. In terminating a job compulsorily, The obstacle recovery processing part 9 updates the status of the job in the job state management table stored in the job state management table storage 11 by the job state Management Department 7 in the state of forced termination, after deleting all the related messages left behind to the message queuing mechanism 4.

[0051]When performing the job accompanied by such a series of processings, the computer resource needed by the resource management department 10 for jobs of the managing server 1 when performing a job is managed per job. At this time, when the resource management department 10 for jobs starts and ends execution of a job, it performs reservation and opening of a required computer resource beforehand. When starting execution of a job, a computer resource is secured, and when it cannot secure, and it waits and execution of a job is able to be secured, specifically, a job is started. On the other hand, when executing each processing program in such a job, the computer resource needed by the resource management department 16 for a program of each execution server 2 when executing each processing program in a job is managed per processing program. At this time, when the resource management department 16 for a program starts and ends execution of a processing program, it performs reservation and opening of a required computer resource beforehand. When starting execution of a processing program, a computer resource is secured, and when it cannot secure, and it waits and execution of a processing program is able to be secured, specifically, execution of a processing program is started. The information about the computer resource needed when executing each processing program in a job or a job is stored in the job definition information storing part 12 per the job unit and processing program.

[0052]Here, in the resource management department 10 for jobs, and the resource management department 16 for a program, the case where it is said that the computer resource concerned is already secured by other jobs and processing programs as a reason a computer resource is not securable is common. In this case, the resource management department 10 for jobs and the resource management department 16 for a program, being concerned -- others -- since the computer resource concerned is not securable until execution of a job or a processing program is completed and a computer resource is opened wide, the job concerned or processing program will be in the state of the waiting for execution. Then, the resource management department 10 for jobs and the resource management department 16 for a program, being concerned -- others -- when execution of a job or a processing program is completed and a computer resource is opened wide, a computer resource is assigned to the processing program in the job in the state of the waiting for execution, or a job, and execution of the processing program in a job or a job is started one by one. It can control that a computer resource required at the time of execution of each processing program in a job or a job is insufficient, and an error occurs by doing in this way and performing exclusive control of a computer resource.

[0053]Thus, according to a 1st embodiment of this invention, the message queuing mechanism 4 for performing asynchronous message communication to each of two or more computers (the managing server 1 and the execution server 2) is formed, Since each processing program is made to cooperate by exchanging the execution demand of each processing program in a job as a message via the message queuing mechanism 4, Processing of a series of the job which consists of two or

more processing programs on two or more computers mutually connected via the network 20 can be performed easily and certainly.

[0054]Namely, since according to a 1st embodiment of this invention the message queuing mechanism 4 is passed and the execution demand of each processing program in a job is carried out, Irrespective of the state of the computer of the partner point, transmission of the message from the managing server 1 to each execution server 2, registration of the message in each execution server 2, execution, and the message from each execution server 2 in the managing server 1 make it wait each other -- etc. -- it can carry out easily and several processing programs with which processing time differs from processing timing etc. can be made to cooperate easily for this reason Even when several computers by which plat forms (OS etc.) differ as the managing server 1 and two or more execution servers 2 are used, data can be appropriately exchanged among these computers (the managing server 1 and the execution server 2). When a message cannot transmit according to the obstacle of the network 20 or execution server 2 grade with the message queuing mechanism 4, Since a message is transmitted when queuing of the message was carried out, a retry is automatically tried when it changed into the state which can communicate, and communication becomes possible, Processing of the data exchanged between computers and a control flow can be guaranteed to the obstacle of the network 20, a computer (the managing server 1 and the execution server 2), etc., and reliability can be maintained.

[0055]The 2nd embodiment, next drawing 6 explain a 2nd embodiment of the batch job control system by this invention. It is the same in a 1st embodiment with which all the computers to which a 2nd embodiment of this invention was connected via the network are execution servers, and indicate others to be to drawing 1 except for a point provided with the function of the managing server which each execution server shows to drawing 1, and abbreviation. In a 2nd embodiment of this invention, identical codes are given to the 1st embodiment and identical parts that are shown in drawing 1, and detailed explanation is omitted.

[0056]As shown in drawing 6, the batch job control system is provided with two or more execution servers (running calculation machine) 21 which execute each processing program in a job. Each execution server 21 has the message queuing mechanism 4 for performing respectively asynchronous message communication, Each processing program is made to cooperate by exchanging an execution demand of a job as a message via the message queuing mechanism 4 between each execution server 21.

[0057]Job definition information 12' about the computer by which each execution server 21 serves as execution sequence of each processing program in a job, and a real destination as an execution demand of a job here, Job state management table 11' showing the stages of progress of the job under execution is exchanged, and each execution server 21 which received the execution demand of a job based on these information achieves the function of both the managing servers 1 and the execution servers 2 which are shown in drawing 1.

[0058]Namely, each execution server 21 has the job execution request reception part 22 which receives job definition information 12' and job state management table 11', An execution demand of the job received by this job execution request reception part 22 is sent to both programs 14 during the executing job control section 6 and execution.

[0059]When an execution demand of a job is received by the job execution request reception part 22, they are specifically sent to the executing job control section 6 by job definition information 12' and job state management table 11', and under control of the executing job control section 6, Job definition information 12' and job state management table 11' are analyzed, and the computer used as the execution sequence of the processing program which is not executed in a job and a real destination is determined. The executing job control section 6 controls execution of the job received by the job execution request reception part 22, Based on the execution sequence of the processing program which is not executed in the job determined by the job definition analyzing parts 8, job definition information 12' and job state management table 11' are transmitted as a message as an



execution demand of the job which contains an unperformed processing program to the applicable execution server 2.

[0060]The processing program which should be executed with the running calculation machine concerned on the other hand among the processing programs in the job received by the job execution request reception part 22 is executed by the program control part 14 during execution.

[0061]At least one of two or more execution servers 21 has the job supply reception part 5 which receives the injection of the job from the outside, An execution demand (job definition information 12' and job state management table 11' are included) of a job can be performed now from the exterior to the execution server 21 used as a starting point.

[0062]Thus, according to a 2nd embodiment of this invention, the message queuing mechanism 4 for performing asynchronous message communication to each of two or more computers (execution server 2) is formed, Since each processing program is made to cooperate by exchanging an execution demand of a job as a message via the message queuing mechanism 4, Processing of a series of the job which consists of two or more processing programs like a 1st embodiment mentioned above on two or more computers (execution server 2) mutually connected via the network 20 can be performed easily and certainly.

[0063]In addition, The 1st mentioned above. And the message queuing mechanism 4 in a 2nd embodiment, the job supply reception part 5, the executing job control section 6, the job state Management Department 7, the job definition analyzing parts 8, the obstacle recovery processing part 9, the resource management department 10 for jobs, the program execution demand accepting part 13, Each can realize the program control part 14, the program execution status management department 15, the resource management department 16 for a program, the job supply part 18, the job state Monitoring Department 19, and the job execution request reception part 22 during execution as a program which operates on a computer. Such a program is recorded on various kinds of recording media, and processing which it was read from the computer and mentioned above is performed.

[0064]As a recording medium used by 1st and 2nd embodiments mentioned above, A magnetic disk, a floppy (registered trademark) disk, a hard disk, As long as computer reading is possible, you can record a program, and what kind of thing may be sufficient as the recording form including optical discs (CD-ROM, CD-R, DVD, etc.), magneto-optical discs (MO etc.), semiconductor memory, etc.? As a recording medium, information transmission media, such as a subcarrier at the time of being transmitted on a network, are included. The recording medium which the recording medium downloaded the program transmitted by what [ not only ] became independent of a computer but LAN, the Internet, etc., and was memorized or stored temporarily is also contained. A recording medium is contained in the concept of the recording medium in this invention further again, also when processing performed by 1st and 2nd embodiments that the number of is not always one and were mentioned above from two or more recording media is realized.

[0065]As a recording medium used by 1st and 2nd embodiments mentioned above, Based on directions of the program installed in the computer from the recording medium, It may be made to realize a part of processing performed by 1st and 2nd embodiments mentioned above by other programs (middleware etc.), such as the operating system (OS) and database management software which are working on a computer, and network software.

[0066]

[Effect of the Invention]As explained above, according to this invention, processing of a series of the job which consists of two or more processing programs on two or more computers mutually connected via the network can be performed easily and certainly.

---

[Translation done.]

\* NOTICES \*

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

---

## DESCRIPTION OF DRAWINGS

---

### [Brief Description of the Drawings]

[Drawing 1]The entire configuration figure showing a 1st embodiment of the batch job control system by this invention.

[Drawing 2]The flow chart for explaining the flow of processing of the whole batch job control system shown in drawing 1.

[Drawing 3]The figure showing an example of job definition information used with the batch job control system shown in drawing 1.

[Drawing 4]The mimetic diagram showing the flow of the job according to the job definition information shown in drawing 3.

[Drawing 5]The figure showing an example of the job state management table used with the batch job control system shown in drawing 1.

[Drawing 6]The entire configuration figure showing a 2nd embodiment of the batch job control system by this invention.

### [Description of Notations]

- 1 Managing server (administrative computer)
- 2, 21 execution servers (running calculation machine)
- 3 Administration terminal
- 4 Message queuing mechanism
- 5 Job supply reception part
- 6 Executing job control section
- 7 Job state Management Department
- 8 Job definition analyzing parts
- 9 Obstacle recovery processing part
- 10 The resource management department for jobs
- 11 Job state management table storage
- 11' job state management table
- 12 Job definition information storing part
- 12' job definition information
- 12a Flow definition master file
- 12b Execution sequence definition file
- 12c Processing program attribute definition file
- 13 Program execution demand accepting part
- 14 It is a program control part during execution.
- 15 Program execution status management department
- 16 The resource management department for a program
- 17 Processing program
- 18 Job supply part



19 Job state Monitoring Department  
20 Network  
22 Job execution request reception part

---

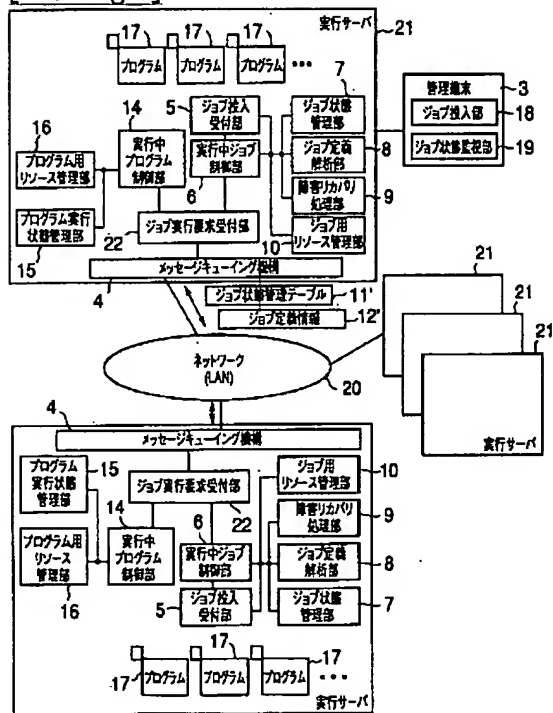
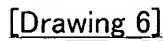
[Translation done.]

JPO and INPIT are not responsible for any damages caused by the use of this translation.

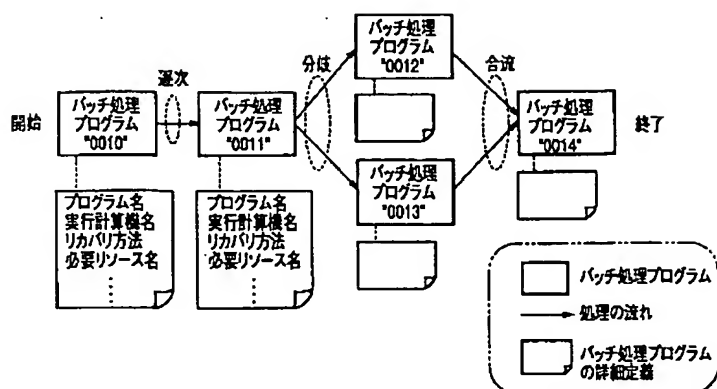
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.\*\*\*\* shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

[Drawing 1]





[Drawing 4]



[Drawing 5]

(ジョブ状態管理テーブル)

11'

ジョブID	業務名	投入日時	投入ユーザ	開始日時	終了日時	現在のステータス	実行中ジョブステップ
00000001	業務A	1999/01/01 22:00:00	admin	1999/01/01 22:10:00	1999/01/01 22:30:00	受付中 JS実行待ち JS実行中 終了(正常) 終了(エラー) タイムアウト発生 強制終了 停止中 リカバリ処理中	0010 0011

プログラムID	処理名	投入日時	実行開始日時	実行終了日時	実行計算機	終了のステータス
0010	処理A	1999/01/01 22:10:00	1999/01/01 22:12:00	1999/01/01 22:30:00	SERVER-A	SS00(正常終了) SS01(エラー) SS02(タイムアウト) nnnnアプリ終了コード

[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-73576

(P2002-73576A)

(43) 公開日 平成14年3月12日 (2002.3.12)

(51) Int. Cl.	識別記号	F I	特許出願公開番号
G 0 6 F 15/16	6 2 0	G 0 6 F 15/16	6 2 0 A 5 B 0 4 5
9/46	3 3 0	9/46	3 3 0 C 5 B 0 9 8
	3 6 0		3 6 0 B
11/00	3 3 0	11/00	3 3 0 A
15/177	6 7 0	15/177	6 7 0 F

審査請求 未請求 請求項の数12 O L (全 12 頁) 最終頁に続く

(21) 出願番号 特願2000-262916 (P2000-262916)

(22) 出願日 平成12年8月31日 (2000.8.31)

(71) 出願人 000003078

株式会社東芝

東京都港区芝浦一丁目1番1号

(72) 発明者 松 尾 朗

東京都府中市東芝町1番地 株式会社東芝  
府中事業所内

(74) 代理人 100064285

弁理士 佐藤 一雄 (外3名)

Pターム(参考) 5B045 BB11 BB23 BB34 BB48 BB49

BB53 CC09 HH02 JJ02 JJ13

JJ42

5B098 AA10 GA03 GB13 CC01 CC16

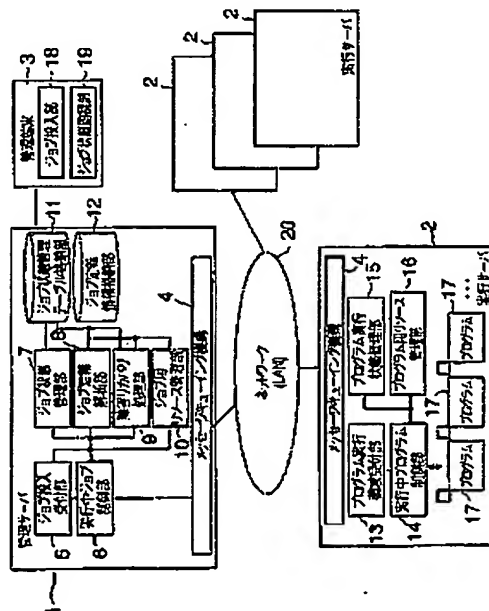
GD01 GD14 JJ03

(54) 【発明の名称】 バッチジョブ制御システム

(57) 【要約】

【課題】 ネットワークを介して互いに接続された複数の計算機上で複数の処理プログラムからなるジョブの一連の処理を容易かつ確実に実行することができるバッチジョブ制御システムを提供する。

【解決手段】 バッチジョブ制御システムは、ジョブ中の各処理プログラムの実行順序および実行先となる計算機を管理する管理サーバ1と、ジョブ中の各処理プログラムを実行する複数の実行サーバ2と、管理サーバ1に接続された管理端末3とを備えている。管理サーバ1と各実行サーバ2とはネットワーク20を介して接続されている。管理サーバ1および各実行サーバ2はそれぞれ非同期のメッセージ通信を行うためのメッセージキューイング機構4を有し、管理サーバ1と各実行サーバ2との間でメッセージキューイング機構4を介してジョブ中の各処理プログラムの実行要求をメッセージとしてやりとりすることにより各処理プログラムを連携させるようになっている。



(2)

特開2002-73576

1

## 【特許請求の範囲】

【請求項1】複数の処理プログラムからなるジョブの実行を制御し、当該各処理プログラムを連携させて一連の処理を実行するバッチジョブ制御システムにおいて、

ジョブ中の各処理プログラムの実行順序および実行先となる計算機を管理する管理計算機と、  
前記管理計算機に接続され、ジョブ中の各処理プログラムを実行する複数の実行計算機とを備え、  
前記管理計算機および前記各実行計算機はそれぞれ非同期のメッセージ通信を行うためのメッセージキューイング手段を有し、前記管理計算機と前記各実行計算機との間で前記メッセージキューイング手段を介してジョブ中の各処理プログラムの実行要求をメッセージとしてやりとりすることにより前記各処理プログラムを連携させることを特徴とするバッチジョブ制御システム。

【請求項2】前記管理計算機は、ジョブの実行要求を受け付けるジョブ投入受付手段と、  
ジョブ中の各処理プログラムの実行順序および実行先となる計算機に関するジョブ定義情報を格納するジョブ定義情報格納手段と、  
前記ジョブ定義情報格納手段に格納されたジョブ定義情報を解析し、前記ジョブ投入受付手段により受け付けられたジョブについて、当該ジョブ中の各処理プログラムの実行順序および実行先となる計算機を決定するジョブ定義解析手段と、

前記ジョブ投入受付手段により受け付けられたジョブの実行を制御し、前記ジョブ定義解析手段により決定された実行順序に基づいて、該当する実行計算機に対して該当する処理プログラムの実行要求をメッセージとして送信する実行中ジョブ制御手段とを有し、

前記各実行計算機は、ジョブ中の各処理プログラムの実行要求のメッセージを受け付けるプログラム実行要求受付手段と、  
前記プログラム実行要求受付手段により受け付けられた実行要求に対応する処理プログラムを実行し、その実行終了通知を前記管理計算機に対してメッセージとして送信する実行中プログラム制御手段とを有することを特徴とする請求項1記載のバッチジョブ制御システム。

【請求項3】前記管理計算機は、実行中のジョブの進行状態を管理するジョブ状態管理手段をさらに有し、  
前記各実行計算機は、実行中の処理プログラムの進行状態を管理するプログラム実行状態管理手段をさらに有することを特徴とする請求項2記載のバッチジョブ制御システム。

【請求項4】前記管理計算機の前記ジョブ定義情報格納手段は、前記ジョブ定義情報の一部として、ジョブまたはジョブ中の各処理プログラムのリカバリ方法に関する情報をさらに格納し、

前記管理計算機は、ジョブの実行中に障害が発生したときに前記ジョブ定義情報格納手段に格納された前記リカ

2

バリ方法に関する情報に基づいてジョブまたはジョブ中の各処理プログラムの復旧処理を行う障害リカバリ処理手段をさらに有することを特徴とする請求項2記載のバッチジョブ制御システム。

【請求項5】前記管理計算機の前記ジョブ定義情報格納手段は、前記ジョブ定義情報の一部として、ジョブを実行する上で必要とされる計算機リソースに関する情報をさらに格納し、

前記管理計算機は、前記ジョブ定義情報格納手段に格納された前記計算機リソースに関する情報に基づいてジョブを実行する上で必要とされる計算機リソースをジョブ単位で管理するジョブ用リソース管理手段をさらに有することを特徴とする請求項2記載のバッチジョブ制御システム。

【請求項6】前記管理計算機の前記ジョブ定義情報格納手段は、前記ジョブ定義情報の一部として、ジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースに関する情報をさらに格納し、

前記管理計算機は、前記各実行計算機に対してジョブ中の各処理プログラムの実行要求とともに前記計算機リソースに関する情報を送信し、

前記各実行計算機は、前記管理計算機から送信された前記計算機リソースに関する情報に基づいてジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースを処理プログラム単位で管理するプログラム用リソース管理手段をさらに有することを特徴とする請求項2記載のバッチジョブ制御システム。

【請求項7】複数の処理プログラムからなるジョブの実行を制御し、当該各処理プログラムを連携させて一連の処理を実行するバッチジョブ制御システムにおいて、

ジョブ中の各処理プログラムを実行する複数の実行計算機を備え、  
前記各実行計算機はそれぞれ非同期のメッセージ通信を行うためのメッセージキューイング手段を有し、前記各実行計算機の間で前記メッセージキューイング手段を介してジョブの実行要求をメッセージとしてやりとりすることにより前記各処理プログラムを連携させることを特徴とするバッチジョブ制御システム。

【請求項8】前記各実行計算機は、ジョブの実行要求として、ジョブ中の各処理プログラムの実行順序および実行先となる計算機に関するジョブ定義情報と実行中のジョブの進行状態に関する情報とを受け付けるジョブ実行要求受付手段と、

前記ジョブ実行要求受付手段により受け付けられたジョブについて、前記ジョブ定義情報と前記実行中のジョブの進行状態に関する情報とを解析し、ジョブ中の未実行の処理プログラムの実行順序および実行先となる計算機を決定するジョブ定義解析手段と、

前記ジョブ投入受付手段により受け付けられたジョブの実行を制御し、前記ジョブ定義解析手段により決定され

(3)

特開2002-73576

3

たジョブ中の未実行の処理プログラムの実行順序に基づいて、該当する実行計算機に対して未実行の処理プログラムを含むジョブの実行要求として前記ジョブ定義情報と前記実行中のジョブの進行状態に関する情報とをメッセージとして送信する実行中ジョブ制御手段と、前記ジョブ実行要求受付手段により受け付けられたジョブ中の処理プログラムのうち、当該実行計算機で実行すべき処理プログラムを実行する実行中プログラム制御手段とを有することを特徴とする請求項7記載のバッチジョブ制御システム。

【請求項9】前記各実行計算機は、実行中のジョブの進行状態を管理するジョブ状態管理手段と、実行中の処理プログラムの進行状態を管理するプログラム実行状態管理手段とをさらに有することを特徴とする請求項8記載のバッチジョブ制御システム。

【請求項10】前記ジョブ定義情報は、ジョブまたはジョブ中の各処理プログラムのリカバリ方法に関する情報をさらに含む。

前記各実行計算機は、ジョブの実行中に障害が発生したときに前記ジョブ定義情報に含まれる前記リカバリ方法に関する情報に基づいてジョブまたはジョブ中の各処理プログラムの復旧処理を行う障害リカバリ処理手段をさらに有することを特徴とする請求項8記載のバッチジョブ制御システム。

【請求項11】前記ジョブ定義情報は、ジョブを実行する上で必要とされる計算機リソースに関する情報をさらに含む。

前記各実行計算機は、前記ジョブ定義情報に含まれる前記計算機リソースに関する情報に基づいてジョブを実行する上で必要とされる計算機リソースをジョブ単位で管理するジョブ用リソース管理手段をさらに有することを特徴とする請求項8記載のバッチジョブ制御システム。

【請求項12】前記ジョブ定義情報は、ジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースに関する情報をさらに含む。

前記各実行計算機は、前記ジョブ定義情報に含まれる前記計算機リソースに関する情報に基づいてジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースを処理プログラム単位で管理するプログラム用リソース管理手段をさらに有することを特徴とする請求項8記載のバッチジョブ制御システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数の処理プログラムからなるジョブの実行を制御するバッチジョブ制御システムに係り、とりわけ、ネットワークを介して互いに接続された複数の計算機の間でジョブ中の各処理プログラムを連携させて一連の処理を実行するバッチジョブ制御システムに関する。

【0002】

4

【従来の技術】従来から、複数の処理プログラムからなるジョブの実行を制御するシステムとして、ジョブ中の複数の処理プログラムを単一の計算機上で順次実行するバッチジョブ制御システムが知られている。

【0003】ところで、近年、ネットワークの普及に伴って複数の計算機が互いに接続された分散システムが広く用いられるようになってきており、このような分散システム上でジョブ中の各処理プログラムを連携させて一連の処理を実行することが望まれている。

【0004】

【発明が解決しようとする課題】しかしながら、上述したような分散システムでは、複数の計算機がネットワークを介して互いに接続されていることから、これらの計算機の間で処理時間や処理タイミング等が異なる複数の処理プログラムを連携させることが困難であるという問題がある。

【0005】また、上述したような分散システムでは、オープン環境下でプラットフォーム（OS等）が異なる複数の計算機が混在して用いられることが多く、これらの計算機の間で適切にデータをやりとりすることが困難であるという問題がある。

【0006】さらに、上述したような分散システムでは、複数の計算機の間で大量のデータの処理および制御を行う必要があり、ネットワークや計算機等の障害に対して、計算機の間でやりとりされるデータの処理および制御の流れを保証して信頼性を保つことが困難であるという問題がある。

【0007】本発明はこのような点を考慮してなされたものであり、ネットワークを介して互いに接続された複数の計算機上で複数の処理プログラムからなるジョブの一連の処理を容易かつ確実に実行することができるバッチジョブ制御システムを提供することを目的とする。

【0008】

【課題を解決するための手段】本発明は、その第1の解決手段として、複数の処理プログラムからなるジョブの実行を制御し、当該各処理プログラムを連携させて一連の処理を実行するバッチジョブ制御システムにおいて、ジョブ中の各処理プログラムの実行順序および実行先となる計算機を管理する管理計算機と、前記管理計算機に接続され、ジョブ中の各処理プログラムを実行する複数の実行計算機とを備え、前記管理計算機および前記各実行計算機はそれぞれ非同期的なメッセージ通信を行うためのメッセージキューイング手段を有し、前記管理計算機と前記各実行計算機との間で前記メッセージキューイング手段を介してジョブ中の各処理プログラムの実行要求をメッセージとしてやりとりすることにより前記各処理プログラムを連携させることを特徴とするバッチジョブ制御システムを提供する。

【0009】なお、上述した第1の解決手段において、前記管理計算機は、ジョブの実行要求を受け付けるジョ

(4)

特開2002-73576

5

6

ブ投入受付手段と、ジョブ中の各処理プログラムの実行順序および実行先となる計算機に関するジョブ定義情報を格納するジョブ定義情報格納手段と、前記ジョブ定義情報格納手段に格納されたジョブ定義情報を解析し、前記ジョブ投入受付手段により受け付けられたジョブについて、当該ジョブ中の各処理プログラムの実行順序および実行先となる計算機を決定するジョブ定義解析手段と、前記ジョブ投入受付手段により受け付けられたジョブの実行を制御し、前記ジョブ定義解析手段により決定された実行順序に基づいて、該当する実行計算機に対して該当する処理プログラムの実行要求をメッセージとして送信する実行中ジョブ制御手段とを有し、前記各実行計算機は、ジョブ中の各処理プログラムの実行要求のメッセージを受け付けるプログラム実行要求受付手段と、前記プログラム実行要求受付手段により受け付けられた実行要求に対応する処理プログラムを実行し、その実行終了通知を前記管理計算機に対してメッセージとして送信する実行中プログラム制御手段とを有することが好ましい。

【0010】また、上述した第1の解決手段において、前記管理計算機は、実行中のジョブの進行状態を管理するジョブ状態管理手段をさらに有し、前記各実行計算機は、実行中の処理プログラムの進行状態を管理するプログラム実行状態管理手段をさらに有することが好ましい。また、前記管理計算機の前記ジョブ定義情報格納手段は、前記ジョブ定義情報の一部として、ジョブまたはジョブ中の各処理プログラムのリカバリ方法に関する情報をさらに格納し、前記管理計算機は、ジョブの実行中に障害が発生したときに前記ジョブ定義情報格納手段に格納された前記リカバリ方法に関する情報に基づいてジョブまたはジョブ中の各処理プログラムの復旧処理を行う障害リカバリ処理手段をさらに有することが好ましい。さらに、前記管理計算機の前記ジョブ定義情報格納手段は、前記ジョブ定義情報の一部として、ジョブを実行する上で必要とされる計算機リソースに関する情報をさらに格納し、前記管理計算機は、前記ジョブ定義情報格納手段に格納された前記計算機リソースに関する情報に基づいてジョブを実行する上で必要とされる計算機リソースをジョブ単位で管理するジョブ用リソース管理手段をさらに有することが好ましい。さらにまた、前記管理計算機の前記ジョブ定義情報格納手段は、前記ジョブ定義情報の一部として、ジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースに関する情報をさらに格納し、前記管理計算機は、前記各実行計算機に対してジョブ中の各処理プログラムの実行要求とともに前記計算機リソースに関する情報を送信し、前記各実行計算機は、前記管理計算機から送信された前記計算機リソースに関する情報に基づいてジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースを処理プログラム単位で管理するプログラム用リソース管理

手段をさらに有することが好ましい。

【0011】本発明は、その第2の解決手段として、複数の処理プログラムからなるジョブの実行を制御し、当該各処理プログラムを連携させて一連の処理を実行するバッチジョブ制御システムにおいて、ジョブ中の各処理プログラムを実行する複数の実行計算機を備え、前記各実行計算機はそれぞれ非同期のメッセージ通信を行うためのメッセージキューイング手段を有し、前記各実行計算機の間で前記メッセージキューイング手段を介してジョブの実行要求をメッセージとしてやりとりすることにより前記各処理プログラムを連携させることを特徴とするバッチジョブ制御システムを提供する。

【0012】なお、上述した第2の解決手段において、前記各実行計算機は、ジョブの実行要求として、ジョブ中の各処理プログラムの実行順序および実行先となる計算機に関するジョブ定義情報と実行中のジョブの進行状態に関する情報とを受け付けるジョブ実行要求受付手段と、前記ジョブ実行要求受付手段により受け付けられたジョブについて、前記ジョブ定義情報と前記実行中のジョブの進行状態に関する情報とを解析し、ジョブ中の未実行の処理プログラムの実行順序および実行先となる計算機を決定するジョブ定義解析手段と、前記ジョブ投入受付手段により受け付けられたジョブの実行を制御し、前記ジョブ定義解析手段により決定されたジョブ中の未実行の処理プログラムの実行順序に基づいて、該当する実行計算機に対して未実行の処理プログラムを含むジョブの実行要求として前記ジョブ定義情報と前記実行中のジョブの進行状態に関する情報とをメッセージとして送信する実行中ジョブ制御手段と、前記ジョブ実行要求受付手段により受け付けられたジョブ中の処理プログラムのうち、当該実行計算機で実行すべき処理プログラムを実行する実行中プログラム制御手段とを有することが好ましい。

【0013】また、上述した第2の解決手段において、前記各実行計算機は、実行中のジョブの進行状態を管理するジョブ状態管理手段と、実行中の処理プログラムの進行状態を管理するプログラム実行状態管理手段とをさらに有することが好ましい。また、前記ジョブ定義情報は、ジョブまたはジョブ中の各処理プログラムのリカバリ方法に関する情報をさらに含み、前記各実行計算機は、ジョブの実行中に障害が発生したときに前記ジョブ定義情報に含まれる前記リカバリ方法に関する情報に基づいてジョブまたはジョブ中の各処理プログラムの復旧処理を行う障害リカバリ処理手段をさらに有することが好ましい。さらに、前記ジョブ定義情報は、ジョブを実行する上で必要とされる計算機リソースに関する情報をさらに含み、前記各実行計算機は、前記ジョブ定義情報に含まれる前記計算機リソースに関する情報に基づいてジョブを実行する上で必要とされる計算機リソースをジョブ単位で管理するジョブ用リソース管理手段をさらに



(5)

特開2002-73576

7

8

有することが好ましい。さらにまた、前記ジョブ定義情報は、ジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースに関する情報をさらに含み、前記各実行計算機は、前記ジョブ定義情報に含まれる前記計算機リソースに関する情報に基づいてジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースを処理プログラム単位で管理するプログラム用リソース管理手段をさらに有することが好ましい。

【0014】本発明の第1および第2の解決手段によれば、複数の計算機のそれぞれに非同期のメッセージ通信を行うためのメッセージキューイング機構4を設け、メッセージキューイング機構4を介してジョブ中の各処理プログラムの実行要求またはジョブの実行要求をメッセージとしてやりとりすることにより各処理プログラムを連携させているので、ネットワークを介して互いに接続された複数の計算機上で複数の処理プログラムからなるジョブの一連の処理を容易かつ確実に実行することができ

【0015】

【発明の実施の形態】以下、図面を参照して本発明の実施の形態について説明する。

【0016】第1の実施の形態

図1乃至図5は本発明によるバッチジョブ制御システムの第1の実施の形態を説明するための図である。

【0017】まず、図1により、本発明の第1の実施の形態に係るバッチジョブ制御システムの全体構成について説明する。図1に示すように、バッチジョブ制御システムは、複数のバッチ処理プログラム（以下単に「処理プログラム」ともいう）からなるジョブの実行を制御し、当該各処理プログラムを連携させて一連の処理を実行するものである。バッチジョブ制御システムは、ジョブ中の各処理プログラムの実行順序および実行先となる計算機を管理する管理サーバ（管理計算機）1と、管理サーバ1に接続され、ジョブ中の各処理プログラムを実行する複数の実行サーバ（実行計算機）2と、管理サーバ1に接続された管理端末3とを備えている。

【0018】図1において、管理サーバ1と各実行サーバ2とはLAN等のネットワーク20を介して接続されている。また、管理サーバ1および各実行サーバ2はそれぞれ非同期のメッセージ通信を行うためのメッセージキューイング機構4を有し、管理サーバ1と各実行サーバ2との間でメッセージキューイング機構4を介してジョブ中の各処理プログラムの実行要求をメッセージとしてやりとりすることにより各処理プログラムを連携させるようになっている。

【0019】ここで、管理サーバ1は、ジョブの実行要求を受け付けるジョブ投入受付部5と、ジョブ中の各処理プログラムの実行順序および実行先となる計算機に関するジョブ定義情報を格納するジョブ定義情報格納部12とを有している。なお、ジョブ定義情報には、ジョブ

中の各処理プログラムの実行順序および実行先となる計算機に関する情報に加えて、ジョブまたはジョブ中の各処理プログラムのリカバリ方法に関する情報と、ジョブまたはジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースに関する情報とが格納されている。なお、ジョブ定義情報の詳細については後述する。

【0020】また、管理サーバ1は、ジョブ定義情報格納部12に格納されたジョブ定義情報を解析し、ジョブ投入受付部5により受け付けられたジョブについて、当該ジョブ中の各処理プログラムの実行順序および実行先となる計算機を決定するジョブ定義解析部8と、ジョブ投入受付部5により受け付けられたジョブの実行を制御し、ジョブ定義解析部8により決定された実行順序に基づいて、該当する実行サーバ2に対して該当する処理プログラムの実行要求をメッセージとして送信する実行中ジョブ制御部6とを有している。ここで、管理サーバ1の実行中ジョブ制御部6は、実行サーバ2に対して処理プログラムの実行要求をメッセージとして送信した後、実行サーバ2から処理プログラムの実行終了通知を待ち、当該実行終了通知を受信した時点で、次の処理プログラムの実行要求をメッセージとして送信し、最後の処理プログラムの実行が終了するまで同様の処理を繰り返すようになっている。なお、管理サーバ1の実行中ジョブ制御部6から送信されたメッセージは管理サーバ1のメッセージキューイング機構4を介して各実行サーバ2に対して送信される。

【0021】さらに、管理サーバ1は、実行中のジョブの進行状態を表すジョブ状態管理テーブルを格納するジョブ状態管理テーブル格納部11と、ジョブ状態管理テーブル格納部11に格納されたジョブ状態管理テーブルを用いて実行中のジョブの進行状態を管理するジョブ状態管理部7とを有している。なお、ジョブ状態管理テーブルには、ジョブ中のどの処理プログラムが現在のどの計算機上で実行されているかという情報の他、既に終了したジョブまたはジョブ中の処理プログラムのステータス情報が格納されており、実行中ジョブ制御部6からの要求に基づいてジョブ状態管理部7により更新されるようになっている。なお、ジョブ状態管理テーブルの詳細については後述する。

【0022】さらにまた、管理サーバ1は、ジョブの実行中に障害が発生したときにジョブ定義情報格納部12に格納されたリカバリ方法に関する情報に基づいてジョブまたはジョブ中の各処理プログラムの復旧処理を行う障害リカバリ処理部9と、ジョブ定義情報格納部12に格納された計算機リソースに関する情報に基づいてジョブを実行する上で必要とされる計算機リソースをジョブ単位で管理するジョブ用リソース管理部10とを有している。なお、ジョブ用リソース管理部10は、メモリやCPU、ディスク容量といった種々の計算機リソースの管理（確保および開放）を行うものであり、実行要求が

(6)

特開2002-73576

9

受け付けられたジョブの実行を開始する際に、ジョブ定義情報格納部12に格納された計算機リソースに関する情報に基づいて必要な計算機リソースの確保を行うようになっている。

【0023】各実行サーバ2は、管理サーバ1の実行中ジョブ制御部6から送信されたジョブ中の各処理プログラムの実行要求のメッセージを受け付けるプログラム実行要求受付部13と、プログラム実行要求受付部13により受け付けられた実行要求に対応する処理プログラムを実行し、その実行終了通知を管理サーバ1に対してメッセージとして送信する実行中プログラム制御部14とを有している。なお、実行中プログラム制御部14は、プログラム実行要求受付部13で受け付けられた実行要求に対応する処理プログラムの起動から終了までの監視を行い、その終了ステータスを管理サーバ1の実行中ジョブ制御部6に対して送信するようになっている。なお、各実行サーバ2の実行中プログラム制御部14から送信されたメッセージは各実行サーバ2のメッセージキューイング機構4を介して管理サーバ1に対して送信される。

【0024】また、各実行サーバ2は、実行中の処理プログラムの進行状態（実行サーバ2上でどの処理プログラムがいくつ同時に実行されているか等）を管理するプログラム実行状態管理部15と、ジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースを処理プログラム単位で管理するプログラム用リソース管理部16とを有している。なお、プログラム用リソース管理部16は、管理サーバ1のジョブ用リソース管理部10と同様に、メモリやCPU、ディスク容量といった種々の計算機リソースの管理（確保および開放）を行うものであり、各実行サーバ2上で処理プログラムの実行を開始する際に、管理サーバ1から引き渡された計算機リソースに関する情報に基づいて必要な計算機リソースの確保を行うようになっている。なお、実行サーバ2には、実際に起動される複数の処理プログラム17がインストールされている。

【0025】なお、管理サーバ1および各実行サーバ2のメッセージキューイング機構4は、管理サーバ1と各実行サーバ2との間でやりとりされるメッセージをキューイングするものであり、管理サーバ1と各実行サーバ2との間でメッセージを非同期にやりとりし、送信先となる実行サーバ2に対してメッセージが確実に届くことを保証するようになっている。すなわち、メッセージキューイング機構4は、ネットワーク20や計算機（管理サーバ1および実行サーバ2）等の障害によりメッセージが送信できないときには、そのメッセージをキューイングし、通信可能な状態になった時点で自動的にメッセージを送信する自動リトライ機能を有している。このため、管理サーバ1または各実行サーバ2は、実行中ジョブ制御部6または実行中プログラム制御部14からメッ

10

セージキューイング機構4に対してメッセージを引き渡した後は、相手先となる計算機に対するメッセージの到着の確認を待つことなく、管理サーバ1または各実行サーバ2の処理を継続することができる。

【0026】管理端末3は、管理サーバ1に接続された利用者（管理者）用の計算機であり、利用者の指示に従って、またはあらかじめ設定された日時に自動的に管理サーバ1に対してジョブの投入（実行要求）を行うジョブ投入部18と、管理サーバ1で管理されている実行中のジョブの進行状態をジョブ状態管理テーブル格納部11に格納されたジョブ状態管理テーブルを参照して監視するジョブ状態監視部19とを有している。

【0027】次に、図1および図2により、このような構成からなる本発明の第1の実施の形態の作用について説明する。図2は図1に示すバッチジョブ制御システムの全体の処理の流れを説明するためのフローチャートである。

【0028】まず、利用者は、実行対象となるジョブごとにジョブ定義情報を作成し、ジョブ定義情報格納部12に格納する。ここで、ジョブ定義情報を作成する作業は、管理端末3上等でGUI等を備えた専用の定義ツールを用いて行うことができる。なお、ジョブ定義情報としては、ジョブ中の各処理プログラムについて、そのプログラム名、実行順序および実行先となる計算機に関する情報の他、ジョブまたはジョブ中の各処理プログラムのリカバリ方法に関する情報や、ジョブまたはジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースに関する情報等が格納されている。

【0029】図3はジョブ定義情報格納部12に格納されたジョブ定義情報の一例を示す図である。

【0030】図3に示すように、ジョブ定義情報12'は、フロー定義マスタファイル12a、実行順序定義ファイル12bおよび処理プログラム属性定義ファイル12cの3つのファイルからなっている。

【0031】このうち、フロー定義マスタファイル12aは、ジョブ定義情報のマスタファイルであり、識別番号（“ID=00000001”）およびジョブ名（“NAME=業務A”）の他、実行順序定義ファイル12bのファイル名（“FILE=F01.net”）が記述されている。

【0032】実行順序定義ファイル12bは、ジョブ中の各処理プログラムの実行順序を定義したものであり、各処理プログラムの識別番号と、それに先行する処理プログラムの識別番号とが複数組記述されている。

【0033】なお、図3に示す実行順序定義ファイル12bでは、識別番号0010の処理プログラムを先頭として、識別番号0010、0011の処理プログラムを逐次実行した後、識別番号0012、0013の処理プログラムを平行して実行し、最後に、識別番号0012、0013の処理プログラムの終了を待ち合わせて識別番号0014の処理プログラムを実行する場合の記述

(7)

特開2002-73576

11

12

例が示されている。

【0034】図4は図3に示す実行順序定義ファイル12bに従ったジョブの流れを示す模式図である。図4に示すように、ジョブの流れは、(1)識別番号0010、0011の処理プログラムのように、先行する処理プログラムの終了を待って次の処理プログラムの実行を開始するタイプ（逐次型）、(2)識別番号0012、0013の処理プログラムのように、同時に複数の処理プログラムを平行して実行するタイプ（分岐型）、(3)識別番号0014の処理プログラムのように、先行する複数の

10

【0035】処理プログラム属性定義ファイル12cは、ジョブ中の各処理プログラムの属性情報を定義したものであり、識別番号（“ID=0010”等）、プログラム名（“NAME=処理A”等）、プログラムのパス（“PATH=C:/.../p10.bat”等）および実行先となる計算機名（“HOST=SERVER-A”等）の他、ジョブまたはジョブ中の各処理プログラムのリカバリ方法の名称（“RECOVERY=リカバリA”等）、およびジョブまたはジョブ中の各

20

【0036】以上のようにしてジョブ定義情報を作成した後、利用者は、管理端末3のジョブ投入部18を用いてジョブの投入（実行要求）を行う。ここで、実行要求を行うことができるジョブは、あらかじめジョブ定義情報格納部12にジョブ定義情報が格納されているジョブのみである。なお、ジョブの実行要求は、利用者の指示によって行う他、あらかじめ起動スケジュール（開始日時および開始対象となるジョブ）を設定しておき、その

30

【0037】図2に示すように、利用者の指示に従って、またはあらかじめ設定された日時に自動的にジョブ投入部18から出されたジョブの実行要求は、管理サーバ1のジョブ投入受付部5で受け付けられ（ステップ101）、実行中ジョブ制御部6へ引き渡される。実行中ジョブ制御部6は、ジョブ定義情報格納部12に対して問い合わせを行い、ジョブ定義情報格納部12に格納されたジョブ定義情報（図3のフロー属性定義マスタファイル12aおよび実行順序定義ファイル12b）を読み込み（ステップ102）、実行要求が出されたジョブの先頭の処理プログラムを抽出するとともにその処理プログラムの実行先となる計算機を決定する（ステップ103）。

【0038】その後、実行中ジョブ制御部6は、該当する処理プログラムの実行要求をメッセージとしてメッセージキューイング機構4へ引き渡し、該当する実行サーバ2に対してメッセージを送信する（ステップ10

50

4）。メッセージキューイング機構4は、実行中ジョブ制御部6から引き渡されたメッセージをキューイングし、該当する実行サーバ2のメッセージキューイング機構4との間でメッセージを非同期にやりとりする。ここで、メッセージキューイング機構4は自動リトライ機能を有しており、ネットワーク20に障害が発生してメッセージが送信できない場合や、送信先となる実行サーバ2がダウンしてメッセージが送信できない場合等において、障害が発生してから障害から復旧するまでの間に一定間隔で自動的にリトライを試み、通信可能な状態になった時点で自動的にメッセージを送信する。なお、ジョブ中の一つの処理プログラムから複数の処理プログラムへ流れが分岐する場合には、管理サーバ1から複数の実行サーバ2に対してメッセージが同報される。また、管理サーバ1と実行サーバ2との間でやりとりされるメッセージは、図3に示す処理プログラム属性ファイル12cに相当するものであり、プログラム名、プログラムのパス、実行先となる計算機名、オプションの情報（リカバリ方法の名称および計算機リソースに関する情報等）等が含まれている。

【0039】ここで、管理サーバ1の実行中ジョブ制御部6から送信された処理プログラムの実行要求のメッセージは該当する実行サーバ2のプログラム実行要求受付部13で受け付けられる（ステップ201）。その後、このようにして受け付けられた処理プログラムの実行要求のメッセージは、実行中プログラム制御部14へ引き渡され、当該メッセージ内に含まれる情報に従って、該当する処理プログラム17が起動される（ステップ202）。なお、実行中プログラム制御部14においては、同時に複数の異なる処理プログラム17を起動することも可能であり、その場合には、それぞれの処理プログラムに対応して複数のスレッドが起動される。

【0040】このようにして処理プログラムが起動されると、実行中プログラム制御部14は、プログラム実行状態管理部15に対して実行中の処理プログラムの登録を行う（ステップ203）。ここで、プログラム実行状態管理部15は、処理プログラムがいくつ同時に実行されているかを管理し、処理負荷の大きなプログラムを同時に多く実行させないように制御する。

【0041】なお、実行中プログラム制御部14は、起動された処理プログラムの実行が終了すると、同様にしてプログラム実行状態管理部15に対して処理プログラムの登録削除要求を行う（ステップ204）。また、実行中プログラム制御部14は、終了した処理プログラムの終了ステータスを取得し、管理サーバ1の実行中ジョブ制御部6に対して処理プログラムの実行終了通知をその終了ステータスとともにメッセージとして送信する（ステップ205）。なお、このメッセージは、実行サーバ2のメッセージキューイング機構4を介して管理サーバ1に対して送信される。

(8)

特開2002-73576

13

【0042】このようにして実行サーバ2から送信された実行終了通知は、管理サーバ1の実行中ジョブ制御部6が受信する(ステップ105)。実行中ジョブ制御部6は、受信した実行終了通知を終了ステータスとともにジョブ状態管理部7へ引き渡し、ジョブ状態管理テーブル格納部11に格納されたジョブ状態管理テーブルを更新する(ステップ106)。

【0043】図5はジョブ状態管理テーブル格納部11に格納されたジョブ状態管理テーブルの一例を示す図である。

【0044】図5に示すように、ジョブ状態管理テーブル11'は、ジョブの進行状態に関する情報(ジョブID、業務名、投入日時、投入ユーザ、終了日時、現在のステータス、実行中ジョブステップ)の他、ジョブ中の各処理プログラムの進行状態に関する情報(プログラムID、処理名、投入日時、実行開始日時、実行終了日時、実行計算機、終了ステータス)が含まれている。

【0045】なお、実行中ジョブ制御部6は、1つの処理プログラムの実行が終了すると、再びジョブ定義解析部8に対して問い合わせを行い、当該ジョブのジョブ定義情報中に次の処理プログラムが存在するかどうかを判断する(ステップ107)。

【0046】ここで、次の処理プログラムが存在する場合には、当該次の処理プログラムを抽出するとともにその処理プログラムの実行先となる計算機を決定し(ステップ108)、上述したステップ104乃至107の処理を繰り返す。なお、複数の処理プログラムから一つの処理プログラムへ流れが合流する場合には、先行する複数の処理プログラムの必要な実行終了通知が全て揃うまで待ち、全ての実行終了通知が揃った時点で、次の処理プログラムの実行要求をメッセージとして送信する。

【0047】なお、当該ジョブ中の最後の処理プログラムが終了した時点で、一つのジョブが完了する(ステップ109)。

【0048】ここで、このような一連の処理を伴うジョブの実行中に障害が発生したときには、障害リカバリ処理部9により、ジョブ定義情報格納部12に格納されたリカバリ方法に関する情報に基づいてジョブまたはジョブ中の各処理プログラムの復旧処理(処理プログラムの再実行や停止、およびジョブの再投入等)を行う。なお、障害リカバリ処理部9は利用者が起動することができる。

【0049】なお、ジョブの実行中に発生する障害としては例えば、実行中の処理プログラムが何らかの原因によりエラーを起こして終了した場合や、ネットワーク20や計算機(管理サーバ1および実行サーバ2)等の障害によって途中で中断したジョブが自動リトライを繰り返したにもかかわらず長時間にわたって停滞してしまう場合等がある。なお、このような場合におけるリカバリ方法はジョブの種類によって異なり、ジョブを再開す

14

る方法として、エラーを起こした処理プログラムから再度実行し直す方法や、エラーを起こした処理プログラムをスキップして再開する方法をとることができる。また、ジョブを再投入して最初の処理プログラムから実行し直す方法や、エラーが発生した時点で強制的にジョブまたは処理プログラムを終了させる方法等をとることができる。なお、それぞれのジョブまたはジョブ中の処理プログラムに対してとられるリカバリ方法は、ジョブ定義情報格納部12に格納されたジョブ定義情報(処理プログラム属性定義ファイル12c等)に記述される。

【0050】ここで、ジョブを再開する場合には、障害リカバリ処理部9は、ジョブ状態管理テーブル格納部11に格納されたジョブ状態管理テーブルからジョブ中の各処理プログラムの進行状態に関する情報を取得し、ジョブ定義情報に記述されたリカバリ方法に従って、再開する処理プログラムを決定した後、実行中ジョブ制御部6に対して処理プログラムの再開要求等を出す。なおこのとき、障害リカバリ処理部9は、メッセージキューイング機構4に残されている関係するメッセージを全て削除する。また、ジョブを再投入する場合には、障害リカバリ処理部9は、メッセージキューイング機構4に残されている関係するメッセージを全て削除し、ジョブ投入受付部5に対して再投入の指示を行う。さらに、ジョブを強制的に終了させる場合には、障害リカバリ処理部9は、メッセージキューイング機構4に残されている関係するメッセージを全て削除した後、ジョブ状態管理部7によりジョブ状態管理テーブル格納部11に格納されたジョブ状態管理テーブル中のジョブのステータスを強制終了の状態に更新する。

【0051】また、このような一連の処理を伴うジョブを実行する際には、管理サーバ1のジョブ用リソース管理部10により、ジョブを実行する上で必要とされる計算機リソースをジョブ単位で管理する。このとき、ジョブ用リソース管理部10は、ジョブの実行を開始および終了する際にあらかじめ必要な計算機リソースの確保および開放を行う。具体的には、ジョブの実行を開始する際には、計算機リソースの確保を行い、確保できない場合にはジョブの実行を待ち合わせ、確保できた時点でジョブを開始する。これに対し、このようなジョブ中の各処理プログラムを実行する際には、各実行サーバ2のプログラム用リソース管理部16により、ジョブ中の各処理プログラムを実行する上で必要とされる計算機リソースを処理プログラム単位で管理する。このとき、プログラム用リソース管理部16は、処理プログラムの実行を開始および終了する際にあらかじめ必要な計算機リソースの確保および開放を行う。具体的には、処理プログラムの実行を開始する際には、計算機リソースの確保を行い、確保できない場合には処理プログラムの実行を待ち合わせ、確保できた時点で処理プログラムの実行を開始する。なお、ジョブまたはジョブ中の各処理プログラム

(9)

特開2002-73576

15

を実行する上で必要とされる計算機リソースに関する情報はジョブ単位および処理プログラム単位でジョブ定義情報格納部12に格納されている。

【0052】ここで、ジョブ用リソース管理部10およびプログラム用リソース管理部16において、計算機リソースを確保できない理由としては、当該計算機リソースが他のジョブや処理プログラムにより既に確保されているという場合が一般的である。この場合には、ジョブ用リソース管理部10およびプログラム用リソース管理部16は、当該他のジョブまたは処理プログラムの実行が終了して計算機リソースが開放されるまで当該計算機リソースを確保することができないので、当該ジョブまたは処理プログラムは実行待ちの状態となる。その後、ジョブ用リソース管理部10およびプログラム用リソース管理部16は、当該他のジョブまたは処理プログラムの実行が終了して計算機リソースが開放された時点で、実行待ちの状態にあるジョブまたはジョブ中の処理プログラムに対して計算機リソースを割り当て、ジョブまたはジョブ中の処理プログラムの実行を順次開始する。なお、このようにして計算機リソースの排他制御を行うことにより、ジョブまたはジョブ中の各処理プログラムの実行時に必要な計算機リソースが不足してエラーが発生することを抑制することができる。

【0053】このように本発明の第1の実施の形態によれば、複数の計算機（管理サーバ1および実行サーバ2）のそれぞれに非同期のメッセージ通信を行うためのメッセージキューイング機構4を設け、メッセージキューイング機構4を介してジョブ中の各処理プログラムの実行要求をメッセージとしてやりとりすることにより各処理プログラムを連携させているので、ネットワーク20を介して互いに接続された複数の計算機上で複数の処理プログラムからなるジョブの一連の処理を容易かつ確実に実行することができる。

【0054】すなわち、本発明の第1の実施の形態によれば、ジョブ中の各処理プログラムの実行要求をメッセージキューイング機構4を介してやりとりしているの、相手先の計算機の状態にかかわらず、管理サーバ1から各実行サーバ2へのメッセージの送信、各実行サーバ2でのメッセージの受け付けおよび実行、および管理サーバ1での各実行サーバ2からのメッセージの待ち合わせ等を容易に行うことができ、このため、処理時間や処理タイミング等が異なる複数の処理プログラムを容易に連携させることができる。また、管理サーバ1および複数の実行サーバ2としてプラットフォーム（OS等）が異なる複数の計算機が用いられる場合でも、これらの計算機（管理サーバ1および実行サーバ2）の間で適切にデータをやりとりすることができる。さらに、メッセージキューイング機構4により、ネットワーク20や実行サーバ2等の障害によりメッセージが送信できないときに、そのメッセージをキューイングし、通信可能な状

16

態になった時点で自動的にリトライを試み、通信可能になった時点でメッセージを送信するので、ネットワーク20や計算機（管理サーバ1および実行サーバ2）等の障害に対して、計算機の間でやりとりされるデータの処理および制御の流れを保証して信頼性を保つことができる。

#### 【0055】第2の実施の形態

次に、図6により、本発明によるバッチジョブ制御システムの第2の実施の形態について説明する。本発明の第2の実施の形態は、ネットワークを介して接続された計算機が全て実行サーバであり、各実行サーバが図1に示す管理サーバの機能を備えている点を除いて、他は図1に示す第1の実施の形態と略同一である。本発明の第2の実施の形態において、図1に示す第1の実施の形態と同一部分には同一符号を付して詳細な説明は省略する。

【0056】図6に示すように、バッチジョブ制御システムは、ジョブ中の各処理プログラムを実行する複数の実行サーバ（実行計算機）21を備えている。各実行サーバ21はそれぞれ非同期のメッセージ通信を行うためのメッセージキューイング機構4を有し、各実行サーバ21の間でメッセージキューイング機構4を介してジョブの実行要求をメッセージとしてやりとりすることにより各処理プログラムを連携させるようになっている。

【0057】ここで、各実行サーバ21は、ジョブの実行要求として、ジョブ中の各処理プログラムの実行順序および実行先となる計算機に関するジョブ定義情報12'と、実行中のジョブの進行状態を表すジョブ状態管理テーブル11'とをやりとりしており、これらの情報に基づいてジョブの実行要求を受け付けた各実行サーバ21が図1に示す管理サーバ1および実行サーバ2の両方の機能を果たすようになっている。

【0058】すなわち、各実行サーバ21は、ジョブ定義情報12'とジョブ状態管理テーブル11'とを受け付けるジョブ実行要求受付部22を有しており、このジョブ実行要求受付部22で受け付けられたジョブの実行要求が実行中ジョブ制御部6および実行中プログラム14の両方に送られるようになっている。

【0059】具体的には、ジョブ実行要求受付部22によりジョブの実行要求が受け付けられると、ジョブ定義情報12'とジョブ状態管理テーブル11'とが実行中ジョブ制御部6に送られ、実行中ジョブ制御部6の制御の下で、ジョブ定義情報12'とジョブ状態管理テーブル11'とを解析し、ジョブ中の未実行の処理プログラムの実行順序および実行先となる計算機を決定する。なお、実行中ジョブ制御部6は、ジョブ実行要求受付部22で受け付けられたジョブの実行を制御し、ジョブ定義解析部8により決定されたジョブ中の未実行の処理プログラムの実行順序に基づいて、該当する実行サーバ2に対して未実行の処理プログラムを含むジョブの実行要求としてジョブ定義情報12'とジョブ状態管理テーブル

(10)

特開2002-73576

17

11' とをメッセージとして送信する。

【0060】一方、ジョブ実行要求受付部22により受け付けられたジョブ中の処理プログラムのうち、当該実行計算機で実行すべき処理プログラムが実行中プログラム制御部14により実行される。

【0061】なお、複数の実行サーバ21のうちの少なくとも一つは外部からのジョブの投入を受け付けるジョブ投入受付部5を有しており、起点となる実行サーバ21に対して外部からジョブの実行要求（ジョブ定義情報12' およびジョブ状態管理テーブル11' を含む）を行

うることができるようにしている。

【0062】このように本発明の第2の実施の形態によれば、複数の計算機（実行サーバ2）のそれぞれに非同期的なメッセージ通信を行うためのメッセージキューイング機構4を設け、メッセージキューイング機構4を介してジョブの実行要求をメッセージとしてやりとりすることにより各処理プログラムを連携させているので、上述した第1の実施の形態と同様に、ネットワーク20を介して互いに接続された複数の計算機（実行サーバ2）上で複数の処理プログラムからなるジョブの一連の処理を容易かつ確実に実行することができる。

【0063】なお、上述した第1および第2の実施の形態におけるメッセージキューイング機構4、ジョブ投入受付部5、実行中ジョブ制御部6、ジョブ状態管理部7、ジョブ定義解析部8、障害リカバリ処理部9、ジョブ用リソース管理部10、プログラム実行要求受付部13、実行中プログラム制御部14、プログラム実行状態管理部15、プログラム用リソース管理部16、ジョブ投入部18、ジョブ状態監視部19およびジョブ実行要求受付部22はいずれも、コンピュータ上で動作するプログラムとして実現することができる。このようなプログラムは、各種の記録媒体に記録され、コンピュータから読み出されて上述したような処理が行われる。

【0064】なお、上述した第1および第2の実施の形態で用いられる記録媒体としては、磁気ディスク、フロッピー（登録商標）ディスク、ハードディスク、光ディスク（CD-ROM、CD-R およびDVD等）、光磁気ディスク（MO等）および半導体メモリ等を含み、プログラムを記録することができ、かつコンピュータ読み取り可能なものであれば、その記録形式はどのようなものであ

【0065】また、上述した第1および第2の実施の形

18

態で用いられる記録媒体としては、記録媒体からコンピュータにインストールされたプログラムの指示に基づき、コンピュータ上で稼動しているオペレーティングシステム（OS）、データベース管理ソフトおよびネットワークソフト等の他のプログラム（ミドルウェア等）により、上述した第1および第2の実施の形態で行われる処理の一部を実現するようにしてもよい。

【0066】

【発明の効果】以上説明したように本発明によれば、ネットワークを介して互いに接続された複数の計算機上で複数の処理プログラムからなるジョブの一連の処理を容易かつ確実に実行することができる。

【図面の簡単な説明】

【図1】本発明によるバッチジョブ制御システムの第1の実施の形態を示す全体構成図。

【図2】図1に示すバッチジョブ制御システムの全体の処理の流れを説明するためのフローチャート。

【図3】図1に示すバッチジョブ制御システムで用いられるジョブ定義情報の一例を示す図。

【図4】図3に示すジョブ定義情報に従ったジョブの流れを示す模式図。

【図5】図1に示すバッチジョブ制御システムで用いられるジョブ状態管理テーブルの一例を示す図。

【図6】本発明によるバッチジョブ制御システムの第2の実施の形態を示す全体構成図。

【符号の説明】

- 1 管理サーバ（管理計算機）
- 2、21 実行サーバ（実行計算機）
- 3 管理端末
- 4 メッセージキューイング機構
- 5 ジョブ投入受付部
- 6 実行中ジョブ制御部
- 7 ジョブ状態管理部
- 8 ジョブ定義解析部
- 9 障害リカバリ処理部
- 10 ジョブ用リソース管理部
- 11 ジョブ状態管理テーブル格納部
- 11' ジョブ状態管理テーブル
- 12 ジョブ定義情報格納部
- 12' ジョブ定義情報
- 12a フロー定義マスタファイル
- 12b 実行順序定義ファイル
- 12c 処理プログラム属性定義ファイル
- 13 プログラム実行要求受付部
- 14 実行中プログラム制御部
- 15 プログラム実行状態管理部
- 16 プログラム用リソース管理部
- 17 処理プログラム
- 18 ジョブ投入部
- 19 ジョブ状態監視部

(11)

特開2002-73576

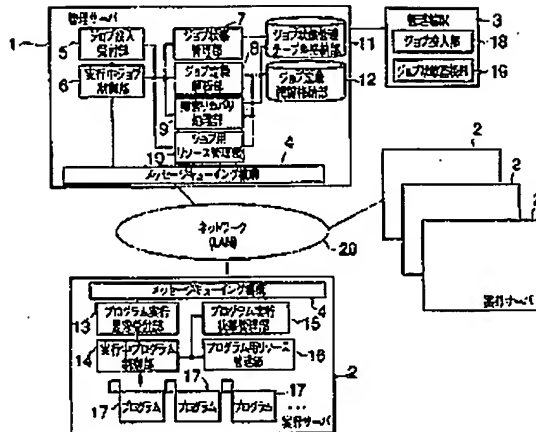
19

20

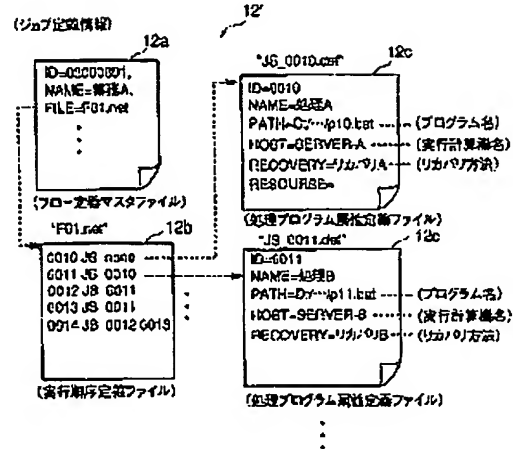
20 ネットワーク

\* \* 22 ジョブ実行要求受付部

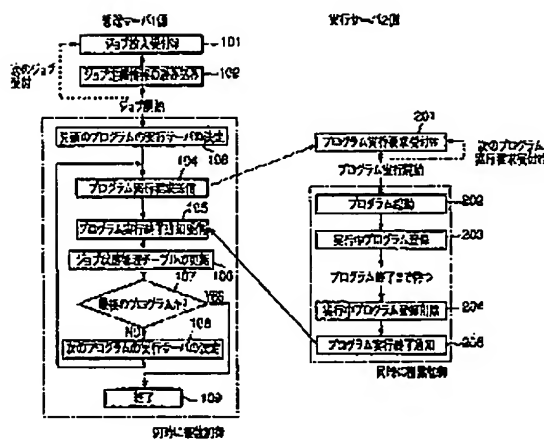
【図1】



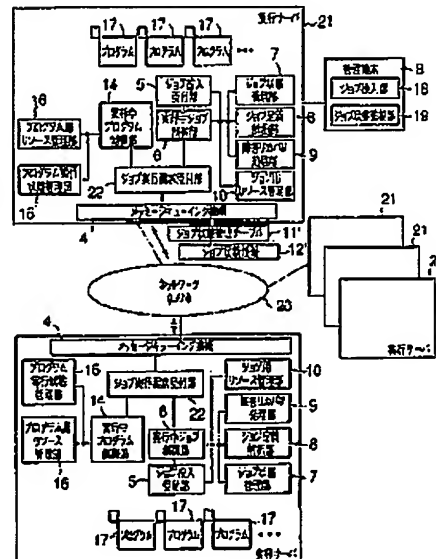
【図3】



【図2】

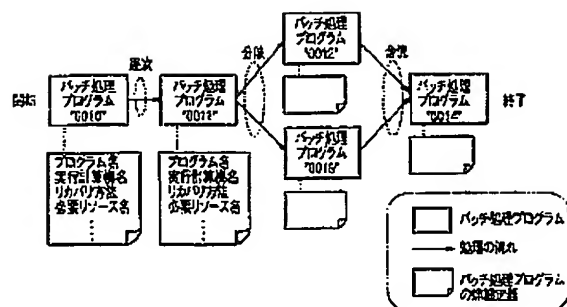


【図6】



(12)

【圖4】



【図5】

(ジョブ状態管理テーブル)

ジョブ状態管理テーブル							
ジョブID	処理名	投入日時	投入ユーザ	開始日時	終了日時	現在のステータス	実行中ジョブステップ
00000001	処理A	1999/01/01 22:00:00	admin	1999/01/01 22:00:00	1999/01/01 22:00:00	実行中 25%完了 85%完了 終了(正常) 終了(エラー) タイムアウト 停止中 実行終了 停止中 ジョブ完了中	0010 0011
プログラムID	処理名	投入日時	実行開始日時	実行終了日時	実行行数	終了のステータス	
0010	処理A	1999/01/01 22:10:00	1999/01/01 22:10:00	1999/01/01 22:00:00	50000000	正常終了 エラー タイムアウト プログラム終了エラー	

フロントページの続き

(51) Int. Cl.  
G 0 6 F 15/177

識別記号  
678

F i  
G O 6 F 15/177

<sup>1</sup>「レポート」(参考)

678C